

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**EV205822556**

**Processing TOC-Less Media Content**

Inventor(s):

Michael J. Polson

David W. Hostetter

Thomas B. Springer

ATTORNEY'S DOCKET NO. MS1-1514US

## **TECHNICAL FIELD**

This invention relates to processing media content that lacks a table-of-contents.

## **BACKGROUND**

With the technological advance of computers and the software that runs on computers, users are now able to enjoy many features, which just a few years ago, did not exist. For example, users can now play various media and multimedia content on their personal or laptop computers, thus providing an improved user experience. For example, most computers today are able to play compact discs (CDs) so that a user can listen to a favorite artist or artists while working on their computer. Additionally, users are now able to “rip” music from a CD to be stored on a computer system in file formats such as .MP3 or .WMA (Windows Media Audio).

As users become more used to advanced features on their computers, such as those mentioned above, their expectations of the various additional innovative features will undoubtedly continue to grow. For example, consider a media player software application that enables a user to play a CD or .MP3 file on their computer. Typical applications allow a user to display, via the use of a mouse, metadata that is associated with the CD by clicking on an appropriate user interface (UI). Such CD metadata typically includes album name, artist name, and song titles. The metadata is not stored on the CD, but is stored in a data repository that can be accessed, for example, over the Internet. The CD includes a structure referred to herein as a table of contents (TOC) that is used to identify the CD. The TOC format, which is defined by the well-known Red Book Audio Standard,

1 specifies the number of tracks, offset of each track on the CD, and the lead out  
2 value. The TOC is used as a key to access the data repository that contains CD  
3 metadata. To further enhance a user's experience, systems have been developed  
4 that allow a user to access additional metadata associated with a particular CD,  
5 including, for example, album art, lists of similar artists, links to websites where  
6 similar music can be purchased, and so on. The additional metadata may also be  
7 accessed based on the Table of Contents.

8 In contrast to a CD, .MP3 files (and other audio files ripped from a CD)  
9 typically do not include a TOC or other unique identifier, but instead include a  
10 defined structure (known as ID3 tags or "attribute tags") for storing textual  
11 metadata associated with the media contained in the file. (Audio files ripped using  
12 Windows Media Player are an exception in that the TOC of the CD is added to the  
13 .WMA file when the file is ripped.) The textual metadata stored in the ID3 or  
14 attribute tags typically includes an artist name, a track name, and an album name.  
15 Unfortunately, however, in many instances, the attribute tags are either not  
16 included, or are manually entered by a user, often resulting in missing or erroneous  
17 metadata. Such media files that do not include a TOC are generally referred to  
18 herein as "TOC-less media".

19 Accordingly, this invention arose out of concerns associated with providing  
20 improved systems and methods for processing TOC-less media content to provide  
21 an improved, rich, and robust user experience.

## 22 23 SUMMARY

24 Methods and systems are described that greatly enhance a user's experience  
25 when playing media content that does not include a table of contents (e.g. MP3

1 and WMA files), herein referred to as "TOC-less media content". One or more  
2 databases, managed by a server, maintain metadata associated with various media.  
3 The metadata can include any type of additional information that can be of interest  
4 to a user or consumer of the media. However, because the media content does not  
5 include a table of contents, identifying associated metadata is not straightforward.  
6 In a described implementation, search criteria may be extracted from attribute tags  
7 that are associated with the media content. In an event that the attribute tags are  
8 blank, search criteria may be extracted by parsing a filename associated with the  
9 TOC-less media content.

10 Because the search criteria is likely to not be complete, several potentially  
11 related sets of metadata may be returned as search results. In an exemplary  
12 implementation, a user can select one of the returned sets of metadata, causing the  
13 metadata to be associated with the TOC-less media in a local media library.  
14 Alternatively, a user can modify the search criteria and perform another search, in  
15 an attempt to identify metadata that should be associated with the TOC-less media.  
16 Another alternative enables a user to manually enter metadata to be associated  
17 with the TOC-less media.

## 18 19 **BRIEF DESCRIPTION OF THE DRAWINGS**

20 Figure 1 is a diagram that illustrates an exemplary environment in which  
21 TOC-less media content can be processed.

22 Figure 2 illustrates an exemplary screen display of track details associated  
23 with TOC-less media content.

24 Figure 3 illustrates the top portion of an exemplary search results screen  
25 display.

1 Figure 4 illustrates the bottom portion of an exemplary search results screen  
2 display.

3 Figure 5 illustrates an exemplary album details screen display.

4 Figure 6 illustrates an exemplary updated track details screen display.

5 Figure 7 illustrates an exemplary refine search screen display.

6 Figure 8 illustrates an exemplary edit track information screen display.

7 Figure 9 is a flow diagram that illustrates exemplary processing of TOC-  
8 less media content.

9 Figure 10 is a flow diagram that illustrates a method for identifying  
10 metadata that may be associated with TOC-less media content.

11 Figure 11 is a flow diagram that illustrates a method for associating  
12 identified metadata with TOC-less media content.

13 Figure 12 is a flow diagram that illustrates a method for enabling a user to  
14 modify search criteria used to identify metadata that may be associated with TOC-  
15 less media content.

16 Figure 13 is a flow diagram that illustrates a method for associating user-  
17 supplied metadata with TOC-less media content.

18 Figure 14 is a block diagram that illustrates selected components of a  
19 computing environment in which TOC-less media content processing can be  
20 implemented.

## 21 **DETAILED DESCRIPTION**

### 22 **Overview**

23 Compact discs (CDs) that are formatted according to the industry standard  
24 Red Book Audio Standard include a table of contents (TOC). The TOC can be  
25

1 used as a key to a database to lookup metadata associated with the media content  
2 stored on the CD. The embodiments described below provide methods and  
3 systems that enable a user or, more accurately, an enabled media player that is  
4 executing on a computing device or client, to access, retrieve, and display for a  
5 user, metadata that is associated with TOC-less media content that is being played  
6 using the media player. A search against a server-based metadata database is  
7 performed, and metadata is returned to the user's computing device, for example  
8 to be stored in a local media library. Furthermore, the enabled media player uses  
9 the retrieved metadata to update attribute tags that are associated with the media  
10 content. In the examples that are given below, the TOC-less media content is  
11 described in the context of content that is embodied in an .MP3 file. It is to be  
12 appreciated and understood that the TOC-less media content can be embodied on  
13 any suitable media, such as in a .WMA file, and that the specific examples  
14 described herein are given for illustrative purposes to assist the reader in  
15 understanding the inventive principles.

16 Various features of the described systems and methods include one or more  
17 databases, client side executable code, and a series of server side processes that  
18 provide for querying and maintaining the databases. One logical organization of  
19 an exemplary system includes the following: (1) a process to extract search criteria  
20 from a TOC-less media file, (2) a process to allow a user to modify the extracted  
21 search criteria, (3) a query process to retrieve information from a database based  
22 on the extracted search criteria, and (4) a process to update a local media library  
23 with metadata that a user identifies as being associated with the TOC-less media  
24 content.

1       The resultant system provides a user with the ability to access additional  
2 metadata and context-sensitive related data (e.g. cover art, performer biographies,  
3 reviews, related performers, where to buy similar items, upcoming concerts, ticket  
4 sales, URLs to other related experiences including buying experiences, and the  
5 like) that may be associated with media content that lacks a table of contents.

### 6       **Exemplary Environment**

7       Figure 1 shows an exemplary environment 100 in which the embodiments  
8 described below can be implemented. The environment 100 includes one or more  
9 client computers—here 101, 102, a network 104, one or more server  
10 computers 106, and one or more databases 108. An exemplary network comprises  
11 the Internet, although any suitable network can be used.

12       In this environment, a user on the client side opens a file containing TOC-  
13 less media content using a media player application on their computer, or  
14 otherwise causes TOC-less media content to be experienced. The media content is  
15 identified by data in embedded attribute tags (if present), the filename, and/or  
16 user-submitted data, which is then used to search a database 108 for associated  
17 metadata. The metadata is then returned to the client via network 104 and  
18 displayed for the user. When the user identifies and selects displayed metadata  
19 that should be associated with the TOC-less media content, the metadata is stored  
20 in association with the media content in a local media library. In an exemplary  
21 implementation, the file containing the media content may also be modified to  
22 include a unique identifier that is also a key in the metadata database. The unique  
23 identifier can then be used at a later time, for example, to access additional  
24 metadata associated with the media content.

1       The description below will provide detailed aspects of the above systems  
2 and various methods that all contribute to a much richer user experience.

### 3       **Wizard**

4       In an exemplary implementation, a series of user interface screens, referred  
5 to herein as a “Wizard”, is provided to assist a user in identifying metadata  
6 associated with media content that lacks a table of contents so that a local media  
7 library can be updated to include metadata associated with the media content. The  
8 Wizard also enables a user to enter metadata associated with the media content in  
9 the event that existing metadata associated with the media content cannot be  
10 identified. The description below presents but one implementation of a Wizard  
11 that can be used to identify, edit, and/or generate metadata associated with TOC-  
12 less media content.

13       Figures 2-8 show select screens of an exemplary Wizard user interface that  
14 can be provided in one implementation. Figure 2 illustrates a track details  
15 screen 200 that is displayed on a client machine when a particular file containing  
16 TOC-less media content is being played using a media player application. In the  
17 illustrated example, a file with the filename “struggling.mp3” is being played. The  
18 file includes data in the ID3 tags that identify the artist and the track name, but no  
19 data in the ID3 tag that identifies the album name. The artist is identified as  
20 “Tricky” (as indicated by the displayed artist name 202) and the track name is  
21 identified as “Strugglin” (as indicated by the displayed track name 204). Because  
22 the album name ID3 tag is blank in the .mp3 file, the media player has no way of  
23 knowing what album the song being played is associated with. Accordingly, no  
24 album art is displayed in the album art area 206. Furthermore, the related  
25 information area 208 contains data that may not be contextually relevant to the



1 music that is currently being played. Because the media player cannot identify an  
2 album associated with the currently playing media content (due to the fact that the  
3 album name ID3 tag is blank), track details screen 200 also includes a selectable  
4 Find Album Info link 210 that enables a user to launch a media identification  
5 process.

6 Figure 3 illustrates the top portion of an exemplary initial results  
7 screen 300, which displays results of an initial search for album information based  
8 on the limited data that is available in the ID3 tags. In the illustrated example, the  
9 track name, "Strugglin" and the artist name "Tricky" are compared to data stored  
10 in a database. Due to the fact that the data in the ID3 tags may have been  
11 manually entered by a user, there is no guarantee that the provided data (e.g., track  
12 name and artist name) is correct or spelled correctly. Accordingly, in an  
13 exemplary implementation, an enhanced text search is performed to increase the  
14 chances of finding data associated with the correct album. Any of a number of  
15 enhanced search techniques may be implemented to reduce the possibility of the  
16 correct album not being returned due to, for example, a spelling, formatting, or  
17 punctuation difference between the data in the database and the data in the ID3  
18 tags. It will be appreciated that such search techniques are well known to those  
19 skilled in the art.

20 Initial results screen 300 includes a track information area 302 in which  
21 known information associated with the current track is displayed. Search results  
22 are displayed in the lower portion of the screen 304. Data associated with each  
23 album that is identified as a potential match is displayed in an area known as a  
24 tile 306. In the illustrated example, each tile displays the album art associated  
25 with the album, the artist name / album name, the track title, the track number, a

1 genre, record label, and release date. In the illustrated exemplary implementation,  
2 when a user moves the cursor 308 over a tile 306, the tile is highlighted, and can  
3 be selected, for example, by the user clicking the mouse.

4 Besides being a selectable entity in and of itself, each tile also includes two  
5 selectable links; an Album Details link 310 and a Buy CDs link 312. In the  
6 illustrated example, when a user selects the Album Details link 310, an album  
7 details page is displayed, which includes a complete list of tracks on the album (an  
8 exemplary album details page is described in more detail below, with reference to  
9 Figure 5). When a user selects the Buy CDs link 312, a music retailer website is  
10 displayed, allowing the user to purchase the identified album or other media  
11 selections.

12 Figure 4 illustrates the bottom portion of the exemplary initial results  
13 screen 300. In the illustrated example, two additional options are provided to the  
14 user in an event that the returned results do not include the album with which the  
15 media content should be associated. A refine search link 402 is provided that,  
16 when selected, enables a user to modify the search criteria in a continued effort to  
17 locate metadata associated with the TOC-less media content. An edit track  
18 information link 404 is provided that, when selected, enables a user to manually  
19 enter metadata to be associated with the current TOC-less media content.

20 Figure 5 illustrates an exemplary album details page 500 that is displayed  
21 when the user selects an album from the results list shown in Figure 3. Album  
22 details page 500 allows a user to view all of the available data associated with the  
23 album to help the user determine whether or not the selected album is indeed the  
24 album that should be associated with the media content currently being played.

1       When a user selects a particular search result tile, as illustrated in Figure 3,  
2       the metadata associated with the selected album is associated with the currently  
3       playing media content in a local media library on the user's client computer  
4       system. After the selected album data is associated with the media content, an  
5       updated track details screen is then displayed.

6       Figure 6 illustrates an example display of an updated track details  
7       screen 600 (which is similar to the track details page 200, illustrated in Figure 2)  
8       that is rendered after a particular album is selected as being associated with the  
9       TOC-less media content that is being played. As can be seen by comparing the  
10      data displayed in track details screen 200 in Figure 2 and updated track details  
11      screen 600 in Figure 6, more complete and context-sensitive metadata is displayed  
12      after the selected album metadata is associated with the media content. For  
13      example, as illustrated in Figure 6, for the illustrated example, artist name 602 is  
14      the only data element displayed in updated track details screen 600 that is the same  
15      as the corresponding data element displayed in track details screen 200 (shown in  
16      Figure 2). The track name 604 has been updated from "Strugglin" to "Strugglin'"  
17      to include the apostrophe at the end of the word; the actual album art is displayed  
18      in album art area 606, where in Figure 2, there was no album art available to be  
19      displayed; and related information area 608 includes data that is more contextually  
20      relevant (e.g., based on the genre associated with the media content) than that  
21      displayed in related information area 208 illustrated in Figure 2. Furthermore,  
22      updated track details screen 600 includes the album name 610, which was not  
23      previously available. Another difference between track details screen 200 and  
24      updated track details screen 600 is that in track detail screen 200, a find album info  
25      link 210 is displayed, allowing the user to search for metadata associated with the

1 current media content; on the other hand, in updated track details screen 600, a  
2 buy CD link 612 is displayed allowing the user to access a retail site through  
3 which the album may be purchased. The buy CD link 612 could not be displayed  
4 on track details screen 200 because there was no information available identifying  
5 the album associated with the current media content. Similarly, the find album  
6 info link 210 does not need to be displayed in updated track details screen 600  
7 because the appropriate album metadata has already been associated with the  
8 media content.

9 Figure 7 illustrates an exemplary display of a refine search screen 700 that  
10 is displayed in response to user selection of the refine search link 402 on initial  
11 results screen 300, as illustrated in Figure 4. Refine search screen 700 displays the  
12 media content filename 702, which is not editable; a track title field 704, which is  
13 initially populated with the value of the track title ID3 tag; an artist name  
14 field 706, which is initially populated with the value of the artist name ID3 tag;  
15 and an album name field 708, which is initially populated with the value of the  
16 album name ID3 tag. A user can correct any erroneous data that is displayed  
17 and/or add any known additional data. Selecting the search button 710 causes  
18 another search to be performed with the updated search criteria, returning another  
19 search results screen, similar to the initial search results screen 300, as illustrated  
20 in Figures 3 and 4.

21 Figure 8 illustrates an exemplary display of an edit track information  
22 screen 800 that is displayed in response to user selection of the edit track  
23 information link 604 on results screen 300, as illustrated in Figure 4. Edit track  
24 information screen 800 displays the media content filename 802, which is not  
25 editable; a track title field 804, which is initially populated with the value of the

1 track title ID3 tag; an artist name field 806, which is initially populated with the  
2 value of the artist name ID3 tag; an album name field 808, which is initially  
3 populated with the value of the album name ID3 tag; a genre field 810, which is  
4 initially blank; and a track number field 812, which is initially blank. A user can  
5 correct any erroneous data that is displayed and/or add any known additional data.  
6 Selecting the save and finish button 814 causes the metadata to be associated with  
7 the current media content in the user's local media library. After the data is saved,  
8 a track details screen similar to those illustrated in Figures 2 and 6 is displayed,  
9 such that the metadata that was entered by the user is displayed and may be used  
10 to identify context sensitive related data, based, for example, on the genre. In the  
11 illustrated example, genre field 810 is implemented as a drop down list to ensure  
12 that the selected genre is recognizable by a system that may be used to identify  
13 context sensitive related data based on the genre.

#### 14 **TOC-Less Media Content Processing**

15 Figure 9 is a flow diagram that describes a method 900 for processing  
16 TOC-less media content performed as a result of user interaction with a Wizard  
17 user interface. The method can be implemented in any suitable hardware,  
18 software, firmware or combination thereof. In the illustrated and described  
19 implementation, the method is implemented in software. This software can reside  
20 on the server side of the system or on the client side of the system. In this  
21 particular example, portions of the software reside on both the server and client  
22 sides of the system. To this extent, Figure 9 is divided into two different  
23 sections—one labeled "Client side" to depict processing that occurs on the client  
24 side, and one labeled "Server side" to depict processing that occurs on the server  
25 side.

1 At block 902, TOC-less media content is being played using a media player  
2 on a client computer system. For example, an .MP3 file is played using a  
3 Windows Media Player application. An exemplary screen display of track details  
4 that may be displayed by such a media player is described above with reference to  
5 Figure 2.

6 At block 904, a media identification process is launched. For example, as  
7 described above with reference to Figure 2, a user selects a find album details link  
8 from a track details screen that displays metadata associated with TOC-less media  
9 content.

10 At block 906, any metadata that is stored in tags associated with the media  
11 content is extracted. For example, if the media content is stored as an .MP3 file,  
12 the data in the ID3 tags is extracted. In the described exemplary implementation,  
13 ID3 tags may store any combination of artist name, track name, and album name,  
14 although it is common for all three of the ID3 tags to be blank. In an exemplary  
15 implementation, in an event that all three ID3 tags are blank, the filename (not  
16 including the extension) may be presumed to be the track name. Alternatively, if  
17 the filename includes a "-", it may be presumed that the filename is of the form  
18 "artist name"- "track name", in which case, the filename is parsed based on the "-"  
19 and the first portion is presumed to be the artist name and the second portion is  
20 presumed to be the track name. The foregoing examples are merely a sample of  
21 possible data extraction techniques. It is contemplated that additional common file  
22 naming practices may be considered, and that additional parsing and data  
23 assignments may be implemented based thereon.

1       At block 908, search parameters are formatted and sent from the client to  
2 the server. In an exemplary implementation, the following data is sent as a search  
3 request:

- 4       • attribute tag values (if available);
- 5       • Media player version;
- 6       • User Locale ID in hex (lcid);
- 7       • Media ID (if available); and
- 8       • Request ID.

9       The attribute tag values are to be used as search parameters for finding  
10 metadata associated with the current media content. The media player version is  
11 used to determine client capabilities and implement response format versioning.  
12 The lcid is passed so that data can be returned in an appropriate format. For  
13 example, in the described exemplary implementation, non-data strings are  
14 displayed by the media player in a language that is associated with the lcid of the  
15 user's browser settings. Furthermore, Unicode input is supported, allowing users  
16 to input metadata in multiple languages, not limited to those supported by ANSI or  
17 ASCII. The media ID is an identifier that is assigned to the media content, and is  
18 used as a database key when looking up metadata associated with the media  
19 content. Typically, the first time a search is performed, and until a result is  
20 selected by a user, the media ID is blank. After an associated album is identified,  
21 the media ID associated with the album may be added to the media file to facilitate  
22 subsequent searches for additional or updated metadata associated with the media  
23 content. The request ID is an identifier that is assigned by the Wizard on the client  
24 side, and is used to later match results returned from the server with the  
25 appropriate search request from which they originated.

1 In an exemplary implementation, search parameters are passed as a  
2 formatted XML file. Other scenarios are also contemplated in which various other  
3 techniques may be used to pass the search parameters from the client to the server.

4 At block 910, the server performs an enhanced text search. In an  
5 exemplary implementation, a query is performed against the enhanced text  
6 matching database 912 using the submitted search terms, resulting in a list of  
7 Media IDs for possible matches. This list of Media IDs is then submitted as a  
8 query against the music metadata database 914, which stores metadata associated  
9 with music albums. The results of the search are then returned to the client.

10 At block 916, the search results are displayed, for example, as illustrated in  
11 Figures 3 and 4.

12 At block 918, the client receives an indication of any one of a number of  
13 user-submitted commands. In an exemplary implementation, the possible user  
14 commands include album details, select match, refine search, manual edit, and buy  
15 CDs.

16 At block 920, in an event that the user submits an album details command  
17 at block 918 (e.g., by selecting the album details link 310 as illustrated in  
18 Figure 3), a client-side display album details process is called, and a media ID  
19 associated with the album for which details were requested is sent to the server.

20 At block 922, a server process requests the album details from the music  
21 metadata database 914, and returns the album metadata to the client for display to  
22 the user. An exemplary display of album details is described above with reference  
23 to Figure 5.

24 At block 924, in an event that the user submits a select match command at  
25 block 918 (e.g., by selecting an album that is displayed in the search results as



1 illustrated in Figure 3), a client side request album metadata process is called and a  
2 media ID associated with the album for which details were requested is sent to the  
3 server. The server side process described above with reference to block 922 is  
4 then performed, and the results are returned to the client.

5 At block 926, the client performs an update media library process to add the  
6 metadata to the user's local media library 928 such that the metadata is associated  
7 with the the TOC-less media content. In an exemplary implementation, the media  
8 ID associated with the selected album is also added, for example, as a binary  
9 GUID, to the TOC-less media content file.

10 At block 930, in an event that the user submits a refine search command at  
11 block 918 (e.g., by selecting the refine search link 402 as illustrated in Figure 4), a  
12 client side refine search criteria process is called. In the described exemplary  
13 implementation, the refine search criteria process causes a screen to be displayed  
14 (e.g., as illustrated in Figure 7) that allows a user to modify the search criteria.  
15 The modified search criteria is then formatted, sent to the server, and processed as  
16 described above with reference to blocks 908 and 910. The search results are then  
17 returned to the client and displayed as described above with reference to  
18 block 916.

19 At block 932, in an event that the user submits a manual edit command at  
20 block 918 (e.g., by selecting the edit track information link 404 as illustrated in  
21 Figure 4), a client side manual edit process is called. In the described exemplary  
22 implementation, the manual edit process causes a screen to be displayed (e.g., as  
23 illustrated in Figure 8) that allows a user to manually enter metadata to be  
24 associated with the media content. When the user submits the manually entered  
25 metadata, the client-side update media library process is called, as described above

1 with reference to block 926 to add the submitted metadata to the user's local  
2 media library 928. In an exemplary implementation, a unique media ID is  
3 associated with the user-submitted data and with the TOC-less media content.

4 At block 934, in an event that the user submits a buy CDs command at  
5 block 918 (e.g., by selecting the buy CDs link 312 as illustrated in Figure 3), a  
6 client-side buy CDs process is launched that causes a screen to be displayed that is  
7 associated with a retailer through which the user can purchase media content.

### 8 **Exemplary Metadata Identification Methods**

9 Figures 10-13 illustrate flow diagrams that describe methods for identifying  
10 metadata to be associated with TOC-less media content. The described methods  
11 can be implemented in any suitable hardware, software, firmware or combination  
12 thereof. In the illustrated and described implementation, the methods are  
13 implemented in software. This software can reside on the server side of the  
14 system or on the client side of the system.

15 Figure 10 illustrates a method 1000 for identifying metadata that may be  
16 associated with TOC-less media content.

17 At block 1002, a file containing TOC-less media content is opened, for  
18 example, using a media player on a client computer system. This corresponds to  
19 block 902, as illustrated in Figure 9.

20 At block 1004, the system receives a request for metadata that may be  
21 associated with the TOC-less media content. This corresponds to block 904, as  
22 illustrated in Figure 9.

23 At block 1006, the system extracts search criteria from the TOC-less media  
24 content file. As described with above, with reference to block 906 of Figure 9,  
25 data may be extracted, for example, from ID3 tags of an MP3 file, from attribute

1 tags of a WMA file, or by parsing a filename associated with the TOC-less media  
2 content file.

3 At block 1008, the system submits the extracted search criteria to a  
4 metadata search system. In the described implementation, the metadata search  
5 system is executed on a server computer system.

6 At block 1010, the system receives search results based on the submitted  
7 search criteria. In an exemplary implementation, metadata results are returned as  
8 XML-formatted data. The received results may range from no results to many  
9 results, where each result is a set of metadata associated with a particular album on  
10 which the TOC-less media content may be a track.

11 At block 1012, the search results are displayed to the user. For example, a  
12 results screen similar to the one illustrated in Figures 3 and 4 may be rendered to  
13 display the search results.

14 Block 1014 represents the entry point to a method 1100 (described below  
15 with reference to Figure 11) for associating with the media content a particular one  
16 of the metadata sets that was returned as a search result.

17 Block 1016 represents the entry point to a method 1200 (described below  
18 with reference to Figure 12) for performing an additional metadata search with  
19 modified search criteria.

20 Block 1018 represents the entry point to a method 1300 (described below  
21 with reference to Figure 13) for associating user-submitted metadata with the  
22 TOC-less media content in an event that the search does not return a metadata set  
23 that is associated with the media content.

24 Figure 11 illustrates a method 1100 for associating metadata with TOC-less  
25 media content.

1       At block 1102, the system receives an indication of a user selection of a  
2 particular set of metadata that was displayed as a search result (as described above  
3 with reference to Figure 10).

4       At block 1104, the system stores the user-selected metadata in a local media  
5 library stored on the user's computer system. In addition, a media ID that is  
6 associated with the metadata is associated with the TOC-less media content. For  
7 example, in an exemplary implementation, a binary GUID that represents the  
8 media ID is added to the file containing the TOC-less media content.  
9 Furthermore, in an exemplary implementation, the attribute tags (e.g. ID3 tags in  
10 an MP3 file) are updated to contain a track name, artist name, and album name as  
11 represented in the selected metadata.

12       Figure 12 illustrates a method 1200 for performing an additional metadata  
13 search with modified search criteria.

14       At block 1202, the system receives an indication of a user selection of an  
15 edit search criteria option. For example, as illustrated in Figure 4, a user may  
16 select the refine your search link 402.

17       At block 1204, the system provides a Wizard user interface screen that  
18 enables the user to modify the search criteria. For example, the screen display  
19 illustrated in Figure 7 may be displayed.

20       At block 1206, the system receives user-submitted modifications to the  
21 search criteria. For example, as illustrated in Figure 7, a user may modify the  
22 search criteria and then select the search button 710.

23       At block 1208, the system submits the modified search criteria to a  
24 metadata search system. As described above, with reference to block 1008, in an  
25

1 exemplary implementation, the metadata search system is executed on a server  
2 computer system.

3 Processing then continues as described above with reference to  
4 blocks 1010-1018 of Figure 10.

5 Figure 13 illustrates a method 1300 for associating user-submitted metadata  
6 with the TOC-less media content in an event that the search does not return a  
7 metadata set that is associated with the media content.

8 At block 1302, the system receives an indication of a user selection of a  
9 manual edit option. For example, as illustrated in Figure 4, a user may select the  
10 edit track information link 404.

11 At block 1304, the system provides a Wizard user interface screen that  
12 enables the user to enter metadata to be associated with the TOC-less media  
13 content. For example, the screen display illustrated in Figure 8 may be displayed.

14 At block 1306, the system receives user-submitted metadata to be  
15 associated with the TOC-less media content. For example, as illustrated in  
16 Figure 8, a user may enter metadata in one or more of the displayed fields and then  
17 select the save and finish button 814.

18 At block 1308, the system stores the user-entered metadata in a local media  
19 library stored on the user's computer system. In addition, a unique media ID is  
20 associated with the metadata and the TOC-less media content. For example, in an  
21 exemplary implementation, a binary GUID that represents the media ID is added  
22 to the file containing the TOC-less media content. Furthermore, in an exemplary  
23 implementation, the attribute tags (e.g. ID3 tags in an MP3 file) are updated to  
24 contain some combination of a track name, artist name, and album name if  
25 included in the user-submitted metadata.

## Exemplary Computer System

Figure 14 illustrates an exemplary computing environment 1400 in which the inventive systems and methods described above can be implemented.

It is to be appreciated that computing environment 1400 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the inventive embodiments described above. Neither should the computing environment 1400 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary computing environment 1400.

The inventive techniques can be operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the inventive techniques include, but are not limited to, personal computers, server computers, thin clients, thick clients, handheld or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

In certain implementations, the inventive techniques can be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The inventive techniques may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In

1 a distributed computing environment, program modules may be located in both  
2 local and remote computer storage media including memory storage devices.

3 In the illustrated example, computing system 1400 includes one or more  
4 processors or processing units 1402, a system memory 1404, and a bus 1406 that  
5 couples various system components including the system memory 1404 to the  
6 processor 1402.

7 Bus 1406 is intended to represent one or more of any of several types of  
8 bus structures, including a memory bus or memory controller, a peripheral bus, an  
9 accelerated graphics port, and a processor or local bus using any of a variety of  
10 bus architectures. By way of example, and not limitation, such architectures  
11 include Industry Standard Architecture (ISA) bus, Micro Channel Architecture  
12 (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association  
13 (VESA) local bus, and Peripheral Component Interconnects (PCI) bus also known  
14 as Mezzanine bus.

15 Computer 1400 typically includes a variety of computer readable media.  
16 Such media may be any available media that is locally and/or remotely accessible  
17 by computer 1400, and it includes both volatile and non-volatile media, removable  
18 and non-removable media.

19 In Figure 14, the system memory 1404 includes computer readable media  
20 in the form of volatile, such as random access memory (RAM) 1410, and/or non-  
21 volatile memory, such as read only memory (ROM) 1408. A basic input/output  
22 system (BIOS) 1412, containing the basic routines that help to transfer information  
23 between elements within computer 1400, such as during start-up, is stored in  
24 ROM 1408. RAM 1410 typically contains data and/or program modules that are  
25

1 immediately accessible to and/or presently being operated on by processing  
2 unit(s) 1402.

3 Computer 1400 may further include other removable/non-removable,  
4 volatile/non-volatile computer storage media. By way of example only, Figure 14  
5 illustrates a hard disk drive 1428 for reading from and writing to a non-removable,  
6 non-volatile magnetic media (not shown and typically called a “hard drive”), a  
7 magnetic disk drive 1430 for reading from and writing to a removable, non-  
8 volatile magnetic disk 1432 (e.g., a “floppy disk”), and an optical disk drive 1434  
9 for reading from or writing to a removable, non-volatile optical disk 1436 such as  
10 a CD-ROM, DVD-ROM or other optical media. The hard disk drive 1428,  
11 magnetic disk drive 1430, and optical disk drive 1434 are each connected to  
12 bus 1406 by one or more interfaces 1426.

13 The drives and their associated computer-readable media provide  
14 nonvolatile storage of computer readable instructions, data structures, program  
15 modules, and other data for computer 1400. Although the exemplary environment  
16 described herein employs a hard disk 1428, a removable magnetic disk 1432 and a  
17 removable optical disk 1436, it should be appreciated by those skilled in the art  
18 that other types of computer readable media which can store data that is accessible  
19 by a computer, such as magnetic cassettes, flash memory cards, digital video disks,  
20 random access memories (RAMs), read only memories (ROM), and the like, may  
21 also be used in the exemplary operating environment.

22 A number of program modules may be stored on the hard disk 1428,  
23 magnetic disk 1432, optical disk 1436, ROM 1408, or RAM 1410, including, by  
24 way of example, and not limitation, an operating system 1414, one or more  
25 application programs 1416 (e.g., media player 1424), other program



1 modules 1418, and program data 1420 (e.g., local media library 1425). Some of  
2 the application programs can be configured to present a user interface (UI) that is  
3 configured to allow a user to interact with the application program in some manner  
4 using some type of input device. This UI is typically a visual display that is  
5 capable of receiving user input and processing that user input in some way. Such a  
6 UI may, for example, include one or more buttons or controls that can be selected  
7 by a user, using an input device such as a mouse. Media player application 1424  
8 can be any suitable media player application that is configured to play any suitable  
9 media so that a user can experience the content that is embodied on the media.  
10 Two exemplary media player applications can include a CD media player  
11 application and a DVD media player application. Local media library 1425 can be  
12 any suitable data storage structure for storing metadata associated with media  
13 content that a user accesses through media player application 1424.

14 Continuing with Figure 14, a user may enter commands and information  
15 into computer 1400 through input devices such as keyboard 1438 and pointing  
16 device 1440 (such as a “mouse”). Other input devices may include audio/video  
17 input device(s) 1453, a microphone, joystick, game pad, satellite dish, serial port,  
18 scanner, or the like (not shown). These and other input devices are connected to  
19 the processing unit(s) 1402 through input interface(s) 1442 that is coupled to  
20 bus 1406, but may be connected by other interface and bus structures, such as a  
21 parallel port, game port, or a universal serial bus (USB).

22 A monitor 1456 or other type of display device is also connected to  
23 bus 1406 via an interface, such as a video adapter 1444. In addition to the  
24 monitor, personal computers typically include other peripheral output devices (not  
25

1 shown), such as speakers and printers, which may be connected through output  
2 peripheral interface 1446.

3 Computer 1400 may operate in a networked environment using logical  
4 connections to one or more remote computers, such as a remote computer 1450.  
5 Remote computer 1450 may include many or all of the elements and features  
6 described herein relative to computer 1400.

7 As shown in Figure 14, computing system 1400 can be communicatively  
8 coupled to remote devices (e.g., remote computer 1450) through a local area  
9 network (LAN) 1451 and/or a general wide area network (WAN) 1452. Such  
10 networking environments are commonplace in offices, enterprise-wide computer  
11 networks, intranets, and the Internet.

12 When used in a LAN networking environment, the computer 1400 is  
13 connected to LAN 1451 through a suitable network interface or adapter 1448.  
14 When used in a WAN networking environment, the computer 1400 typically  
15 includes a modem 1454 or other means for establishing communications over the  
16 WAN 1452. The modem 1454, which may be internal or external, and may be  
17 connected to the system bus 1406 via the input interface 1442, or other appropriate  
18 mechanism.

19 In a networked environment, program modules depicted relative to the  
20 personal computer 1400, or portions thereof, may be stored in a remote memory  
21 storage device. By way of example, and not limitation, Figure 14 illustrates remote  
22 application programs 1416 as residing on a memory device of remote  
23 computer 1450. It will be appreciated that the network connections shown and  
24 described are exemplary and other means of establishing a communications link  
25 between the computers may be used.

## Conclusion

The systems and methods described above can greatly enhance the user's media experience when they play TOC-less media content in an enabled player. A robust collection of metadata is available for provision to the user through the use of a data extraction and search process that enables a user to locate metadata associated with the TOC-less media content.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.